**RELAZIONE**

**Obiettivo del progetto:**

Realizzare un’applicazione client-server che consenta all’utente di consultare informazioni di interesse relative a rifiuti prodotti dalla regione Veneto presenti in un file csv.

**Descrizione del progetto**

1. Progettazione del server: leggere i dati dal file CSV e memorizzarli una struttura dati, creazione di una interfaccia per la comunicazione con un client tramite socket.
2. Progettazione di un’applicazione client: permettere all’utente di richiedere dati specifici al server, visualizzare i dati ottenuti.
3. Progettazione di un protocollo di comunicazione: permettere a client e server di comunicare in maniera semplice.

**Scelte progettuali:**

**Server:**

* Utilizzo del protocollo TCP per il trasporto dati in quanto questo è più semplice da gestire con il linguaggio java.
* Utilizzo dei Thread per gestire più connessioni. Quando viene fatta una richiesta di comunicazione questa richiesta sarà gestita da un nuovo thread. Ogni Thread gestisce un menù che permetterà di eseguire l’opzione scelta dal client.
* Implementazione della classe “GestoreDati” nella quale viene letto il file csv tramite l’utilizzo di uno scanner; il dato ottenuto, con i suoi campi, viene memorizzato in un ArrayList. La classe implementa anche tutti i metodi che permettono la rielaborazione dei dati. Questi metodi devono essere accessibili da tutti i thread e quindi da tutti i client connessi al server.
* Grazie alla struttura dati ArrayList viene garantita una gestione dei dati memorizzati piuttosto semplice anche se non vi è la garanzia che i dati memorizzati non siano duplicati. L’ArrayList è la struttura dati perfetta in quanto, in questo caso, non è necessario garantire l’univocità dei dati.
* Il server restituirà il dato richiesto tramite una stringa rielaborata.
* Utilizzo dei codici per associare univocamente un’opzione ad un metodo della classe “GestoreDati”.
* Utilizzo di JavaDOC per la descrizione di ciascun metodo sviluppato.

**Client:**

* Utilizzo del socket java.
* Utilizzo dell’interfaccia grafica (Java Swing) per mostrare le opzioni disponibili all’utente.
* Creazione di una finestra (Jframe) per visualizzare il menù.
* Quando l’utente sceglie l’opzione il client invierà al server un intero (il codice)che servirà ad identificare univocamente il metodo che permette di eseguire tale operazione.
* Creazione di un’altra finestra nel momento in cui l’utente seleziona l’opzione. Questa finestra viene gestita da un Thread in modo tale da permettere all’utente di fare più richieste al server anche allo stesso tempo.
* Utilizzo di JTextField e JcomboBox per permettere all’utente di inserire i filtri per la ricerca sul file csv.
* Visualizzazione della stringa ricevuta dal server tramite Jlabel.
* Utilizzo di JavaDOC per la descrizione di ciascun metodo sviluppato

**Protocollo di comunicazione utilizzato:**

Per comodità ho scelto di associare ad ogni operazione un codice identificativo univoco in modo tale da:

* Diminuire lo scambio di dati tra client e server
* Semplificare l’implementazione dell’interfaccia grafica
* Semplificare l’identificazione da parte del server dell’opzione scelta dall’utente

**Descrizione:**

1. Connessione:

* Il server si mette in attesa
* Il client effettua la richiesta di connessione

1. Selezione dell’opzione:

* Il client legge e invia l’opzione scelta dall’utente
* Il client permette l’inserimento di eventuali filtri e li invia al server
* Il server, in base all’operazione scelta, si mette in ascolto per ricevere i filtri o esegue direttamente la ricerca

1. Esecuzione della ricerca e invio dei dati rielaborati:

* Il server effettua la ricerca sul file in base agli eventuali filtri ricevuti
* Il server invia al client una stringa contenente i dati rielaborati

1. Visualizzazione dei dati:

* Il client riceve la stringa rielaborata e la visualizza tramite interfaccia grafica.

ES: (codice 1)

Richiesta del client: 1 (per ottenere una riga del file)

Richiesta del client: n (ottenere la riga n del file)

Risposta del server: toString() del dato o “ERROR: riga non trovata” (messaggio di errore)

Visualizzazione dei dati da parte del server

**Opzioni disponibili all’utente:**

* Riga del flie csv: Il client permette l’inserimento di un intero corrispondente al numero della riga, ed il server restituisce il toString() del dato corrispondente alla riga selezionata. (tramite metodo “getRiga()” istanziato nella classe “GestoreDati”) **Codice 1.**
* Provincia che ha prodotto più rifiuti in un determinato anno: Il server richiede l’anno, il client permette l’inserimento, il server analizza i dati memorizzati (tramite metodo “provinciaRifiutiAnno()” istanziato nella classe “GestoreDati”) e invia la stringa “La provincia che a prodotto più rifiuti nell’anno x è y”. **Codice 2.**
* Quantità di rifiuti prodotti per tipo dalla regione: Dopo la richiesta del client il server invia la stringa “La regione Veneto ha prodotto x rifiuti pericolosi e y rifiuti speciali” (tramite metodo “rifiutiProdotti()” istanziato nella classe “GestoreDati”). **Codice 3.**
* Quantità di rifiuti prodotti per tipo in un anno:Il server richiede l’anno, il client permette l’inserimento, il server analizza i dati memorizzati (tramite metodo “rifiutiProdottiAnno()” istanziato nella classe “GestoreDati”) e invia la stringa “Nell’anno x sono stati prodotti y rifiuti pericolosi e z rifiuti speciali”. **Codice 4.**
* Quantità di rifiuti prodotti per tipo da una provincia:Il server richiede la provincia, il client permette l’inserimento, il server analizza i dati memorizzati (tramite metodo “rifiutiProdottiProvincia()” istanziato nella classe “GestoreDati”) e invia la stringa “La provincia x nell’anno y ha prodotto z rifiuti pericolosi e k rifiuti speciali ”. **Codice 5.**
* Anno in cui sono stati prodotti più rifiuti: Dopo la richiesta del client il server invia la Stringa “La regione Veneto ha prodotto più rifiuti nel x” (tramite metodo “annoRifiuti()” istanziato nella classe “GestoreDati”). **Codice 6.**
* Provincia che ha prodotto più rifiuti: Dopo la richiesta del client il server invia la Stringa “La provincia che ha prodotto più rifiuti è : x” (tramite metodo “provinciaRifiuti() istanziato nella classe “GestoreDati”). **Codice 7.**